

МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ДАННЫХ ДЛЯ ХРАНЕНИЯ РЕЗУЛЬТАТОВ ИЗМЕРЕНИЙ В СИСТЕМЕ ТЕСТИРОВАНИЯ ЭФФЕКТИВНОСТИ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

Введение

В последние годы наблюдается активизация использования индивидуальных измерительных приборов, способных обмениваться данными с компьютером, – таких как пульсометры и развлекательные энцефалографы. Хотя эти устройства и не ориентированы на научные цели, они могут предоставить достаточно точные данные о физическом и эмоциональном состоянии пользователя. Ниже представляется результат авторской разработки, нацеленной на создание программно-аппаратной системы приборной оценки удобства использования ПО персонального компьютера на основе анализа таких данных.

Основная часть ПО написана на C++ и Qt, в настоящее время скомпилирована для двух платформ: GNU/Linux и MS Windows.

1. Выбор СУБД

В ходе разработки приложения рассматривались три свободно распространяемые системы управления базами данных: мощная встраиваемая система **SQLite**, самая распространённая на текущий момент СУБД **MySQL**, и наиболее развитая серверная СУБД **PostgreSQL**.

SQLite - легко встраиваемая в приложения база данных, базируется на файлах, а потому она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с этой СУБД обращения происходят напрямую к файлам (в эти файлы хранятся данные), вместо портов и сокетов сетевых СУБД. Именно поэтому SQLite очень быстрая, а также мощная благодаря технологиям обслуживающих библиотек. К ее минусам можно отнести отсутствие многопользовательского доступа и низкую производительность при записи больших объемов данных (разрешен только один процесс записи в промежуток времени).

PostgreSQL является самой профессиональной из всех трех рассмотренных нами СУБД. Она свободно распространяемая и максимально соответствует стандартам SQL. От других СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного к базам данных наряду с классическим реляционным подходом. Благодаря мощным технологиям PostgreSQL имеет высокую производительность и масштабируемость. Параллельность достигнута не за счет блокировки операций чтения, а благодаря реализации управления многовариантным параллелизмом (MVCC). PostgreSQL очень легко расширять собственными процедурами и модулями расширения. К минусам можно отнести невысокую производительность при простых операциях чтения и не самую высокую популярность.

MySQL - самая распространенная полноценная серверная СУБД. MySQL очень функциональная, свободно распространяемая СУБД. Несмотря на то, что в ней не реализован весь функционал языка SQL, MySQL предлагает довольно много инструментов для разработки приложений. Так как это серверная СУБД, приложения для доступа к данным, в отличие от SQLite, работают со службами MySQL. Также среди преимуществ можно отметить простоту установки и использования, большое количество функций, обеспечивающих безопасность по умолчанию, то что MySQL легко работает с большими объемами данных и легко масштабируется, а также значительную производительность (достигаемую в т. ч. за счет упрощения некоторых стандартов).

На основе вышеизложенного была выбрана реляционная СУБД MySQL. В качестве подсистемы таблиц СУБД был выбран вариант InnoDB, который входит во все стандартные сборки для различных операционных систем. Основным отличием InnoDB от других подсистем низкого уровня MySQL является наличие механизма транзакций и внешних ключей.

2. Структура хранения данных

При проектировании базы данных была разработана структура таблиц и связей, представленная на рисунке 1. Данная база является реляционной и приведена к третьей нормальной форме схем отношений, что является достаточным в большинстве случаев и не приводит к избыточности таблиц, что в свою очередь снижает скорость обработки данных и усложняет администрирование.

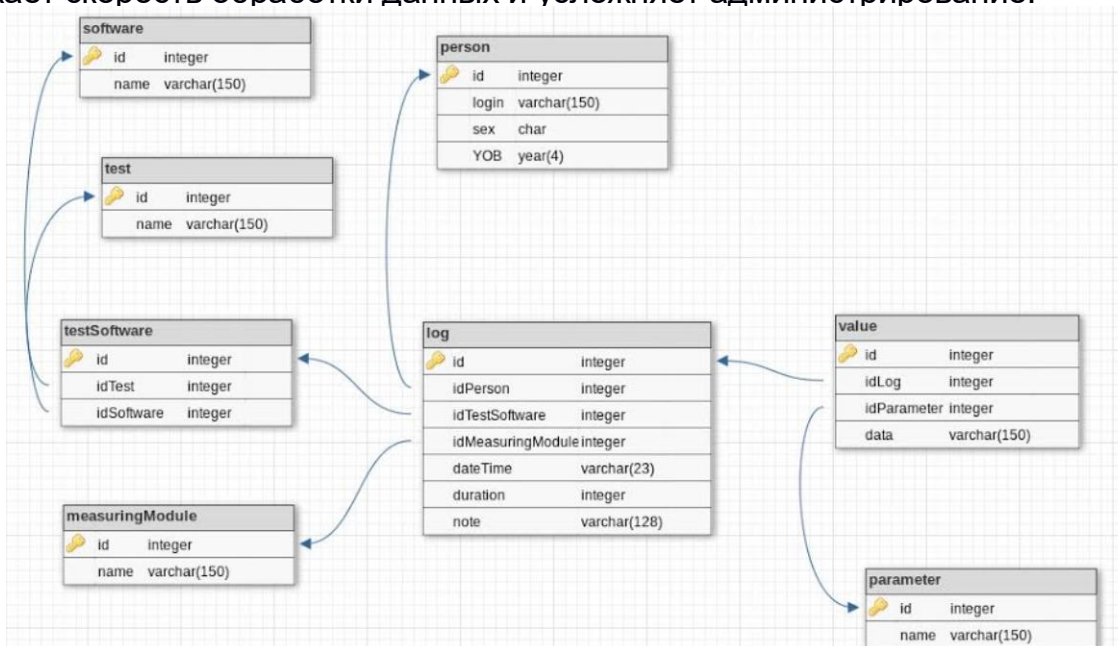


Рисунок 1 – Структура хранения данных

Связи между таблицами организованы внешними ключами типа «один ко многим». Это позволяет организовать каскадное удаление и изменение данных в дочерних таблицах.

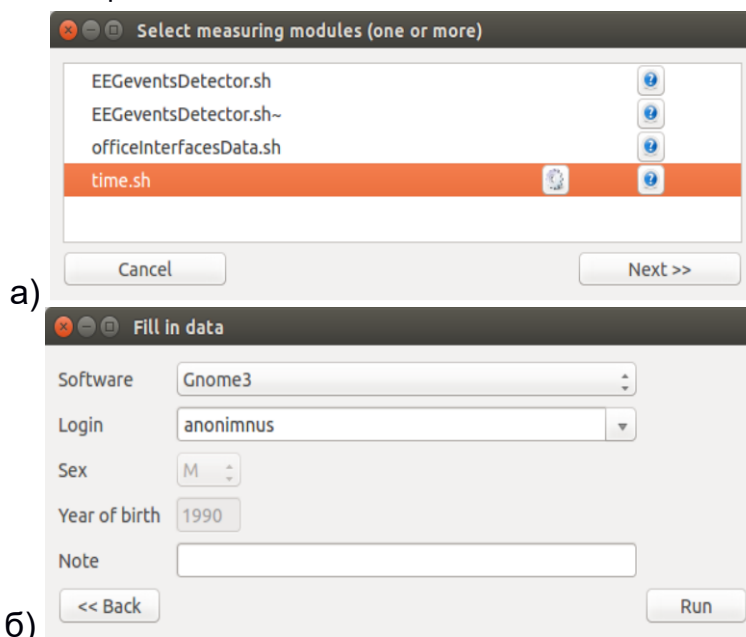


Рисунок 2 – Выбор измерительных модулей (а) и заполнение данных об эксперименте (б)

База данных хранит результаты тестирования эргономики ПО, собираемые одновременно с нескольких компьютеров. Для этого на каждом компьютере, участвующем в тестировании, должна быть запущена копия клиентского приложения. В задачи этого приложения входит сбор данных от определенных измерительных модулей (соответствующих надетым на пользователя датчикам) и запуск выбранного пользователем модуля тестирования ПО. Список доступных измерительных модулей (рис. 2-а) и модулей тестирования определяется клиентским приложением автоматически; по окончании сеанса тестирования полученные от измерительных модулей данные заносятся в пару таблиц `parameter` и `value` (по причине того, что набор измеряемых модулями параметров заранее не известен). Сведения о выбранных измерительных модулях и модуле тестирования, а также информация о пользователе и ПО, с которым он работал в ходе данного эксперимента (рис. 2-б), помещаются в таблицу `log` вместе с отметкой времени, длительностью эксперимента и опциональным текстовым комментарием. Если какая-либо из перечисленных сущностей ранее не участвовала в экспериментах и потому отсутствует в базе данных, сведения о ней помещаются в соответствующую таблицу (`person` для пользователей, `software` для тестируемого ПО, `test` для тестового модуля, и `measuringModule` для измерительных модулей). Также с целью нормализации была введена промежуточная сущность `testSoftware`, связывающая между собой тестируемое ПО и тестовый модуль, который его запускает и контролирует процесс работы.

В результате работы серверная база данных накапливает значительное количество измерений. Программная оболочка, позволяющая выполнять к ним доступ, снабжена гибкой системой динамической фильтрации данных (рис. 3).

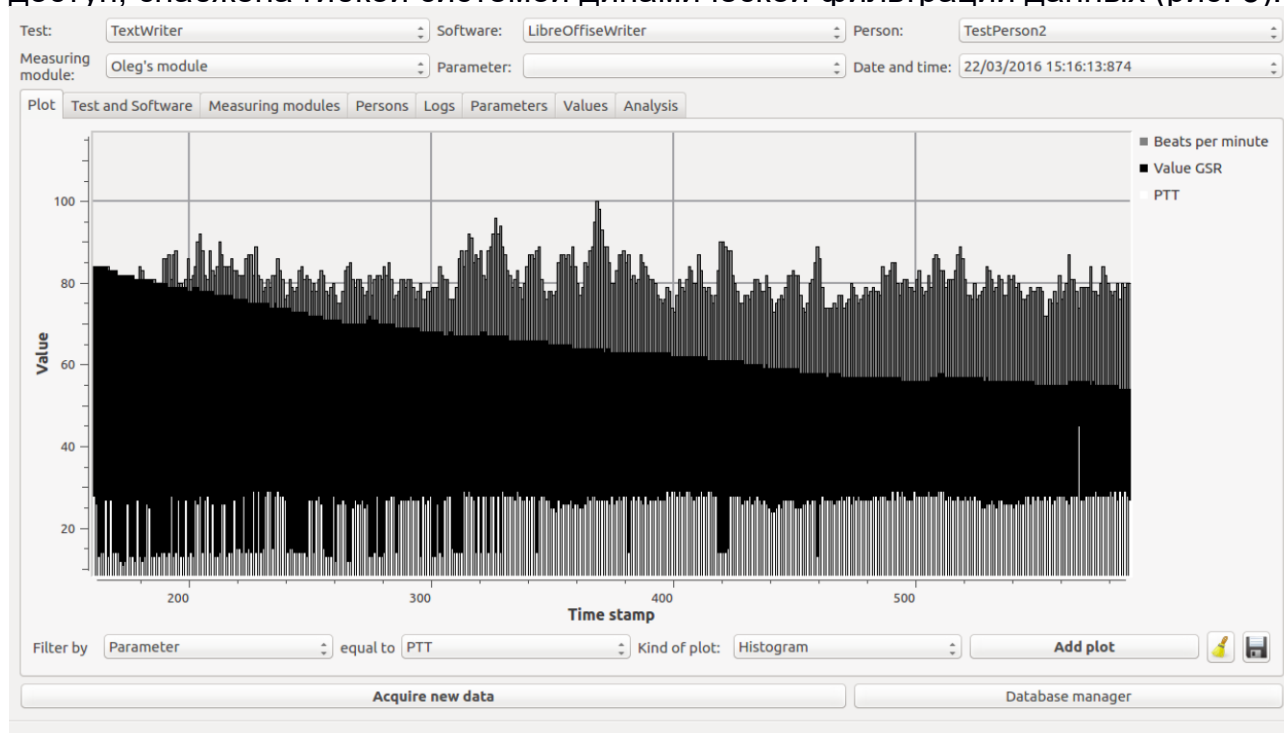


Рисунок 3 – Диалог доступа к результатам измерений

Фильтрация организована с помощью объемного SQL-запроса глубокой степени вложенности, с подзапросами, охватывающими все сущности базы данных. Выполнение такого запроса ресурсоемко, и для ускорения его обработки была реализована схема индексирования, представленная в следующей таблице, также иллюстрирующей преимущества выбранного решения по сравнению с вариантом, использующим только первичные ключи связей таблиц.

Таблица 1 – Сравнение эффективности обработки запроса в зависимости от использования индексов базы данных

Таблица БД	Без индексирования на стороне сервера			С индексированием на стороне сервера		
	Число записей, пройденных при выборке	Доступные к использованию ключи	Используемые ключи	Число записей при выборке	Доступные к использованию ключи	Используемые ключи
person	1	PRIMARY	NULL	1	PRIMARY	NULL
test	1	PRIMARY	NULL	1	PRIMARY	NULL
software	1	PRIMARY	NULL	1	PRIMARY	NULL
measuringModule	1	PRIMARY	NULL	1	PRIMARY	NULL
person	1	PRIMARY	NULL	1	PRIMARY	NULL
test	1	PRIMARY	NULL	1	PRIMARY	NULL
software	1	PRIMARY	NULL	1	PRIMARY	NULL
measuringModule	1	PRIMARY	NULL	1	PRIMARY	NULL
log	1	PRIMARY	PRIMARY	6	PRIMARY, idPerson, idTest, idSoftware, idMeasuringModule	idPerson
parameter	156	NULL	NULL	6	idLog	idLog
log	1	PRIMARY	PRIMARY	13	PRIMARY, idPerson, idTest, idSoftware, idMeasuringModule	NULL
value	102753	NULL	NULL	313	logColumn	logColumn

Список цитированных источников

1. Костюк, Д.А. Инструментальная оценка состояния пользователя в задаче сравнения интерфейсов офисных приложений / Д.А. Костюк, О.О. Латий, А.А. Маркина // XII конференция разработчиков свободных программ: тезисы докладов. – Калуга, 16-18 октября 2015г.М.: Альт Линукс, 2015. – С.8–12

2. Журавский, В.В. Программно-аппаратная система для сравнительных исследований эргономики программного обеспечения / В.В. Журавский, Д.А. Костюк, О.О. Латий, А.А. Маркина // Информационные технологии и системы 2015 (ИТС 2015): материалы Международной научной конференции. Минск, БГУИР, 29 октября 2015 г. – С. 252–253.

3. Латий, О.О. Инструментальная оценка эффективности пользовательского взаимодействия с интерфейсом офисных пакетов // Современные проблемы математики и вычислительной техники : сборник материалов IX Республиканской научной конференции молодых ученых и студентов, Брест, 19-21 ноября 2015 г. / Брестский государственный технический университет ; редкол.: В.С. Рубанов (гл.ред.) [и др.]. - Брест: БрГТУ, 2015. - С. 47-49.